

基于大语言模型的电商金融分析系统

计金（双） 刘宇晨 20002515

摘要：随着电商金融的快速发展，个性化推荐系统成为提升用户体验和增加用户粘性的关键技术。为了更好地理解用户的需求和偏好，本文设计了一个名为 ref2label 的大语言模型，旨在将用户评论精简为商品标签。该模型采用了 transformer 架构，通过 Kmeans 聚类算法，实现了一个能够不断自我更新的推荐系统。系统通过分析和处理源源不断的用户评论，自动转换成 n 个最能代表商品特点的标签。这些标签不仅反映了用户的真实诉求，而且通过 fine-tuning 模型和词嵌入技术的结合，能够为用户推荐最适合他们的商品。实验结果表明，ref2label 模型能够有效提高推荐系统的准确率和用户满意度，为电商平台提供了一种新的用户体验优化手段。

关键词：transformer 模型、K-means 聚类算法、fine-tuning 模型、词嵌入技术

1 研究背景

1.1 选题背景

近年来，伴随着深度学习与大语言模型的飞速发展，我们见证了自然语言处理（NLP）领域的一次又一次突破。自从 Transformer 模型问世以来，在大语言模型领域涌现出了一系列高度可用的模型，包括 GPT 系列和 BERT 及其变体等。这些模型在自然语言处理任务中取得了令人瞩目的成果，并推动了该领域的创新发展。这些进步不仅推动了机器学习理论的深入发展，也为实际应用提供了前所未有的可能性。特别是在文本生成、情感分析、机器翻译、语言理解等方面，大语言模型展现出了惊人的能力，极大地丰富了人们对语言智能化处理的认识和期待。

随着模型性能的不不断提升，其应用范围也在不断扩大。在电子商务、金融科技、股市情感预测、社交媒体分析等多个领域，大语言模型已成为解决关键业务问题、提升服务质量、增强用户体验的重要工具。尤其是在电子商务领域，通过深入分析用户评论、反馈和行为数据，大语言模型能够帮助企业更好地理解用户需求，实现更加个性化的商品推荐，从而提升用户满意度和购物体验。

1.2 研究意义

在当前的电子商务领域，推荐系统的发展已达到了一个高度成熟的阶段，能够利用用户的购买历史、浏览习惯等数据，通过复杂的算法为用户推荐他们可能感兴趣的商品。这种基于历史数据的推荐技术，虽然在很大程度上优化了用户的购物体验，增加了平台的销售额，但仍存在一些不可忽视的局限性。特别是当用户面对具有模糊性或多样化需

求的购物决策时，现有的推荐系统往往难以准确捕捉到用户的真实意图，导致推荐结果可能与用户期望存在偏差。

随着电子商务与金融科技的融合发展，用户的购物行为和消费需求变得更加多元和复杂。用户不仅仅在寻找特定的商品，更希望通过购物过程获得满足个性化需求的体验。在这样的背景下，传统推荐系统的局限性就显得尤为突出。因此，探索新的推荐技术，以更准确地理解和满足用户的个性化需求，成为了电商平台提升竞争力的关键。

本文提出的研究，通过引入生成式大语言模型来优化推荐系统，正是基于以上背景和考虑。该模型可以接受用户以自然语言表达的购买诉求，通过智能分析和处理这些文本数据，生成与用户需求高度相关的商品特征标签。这些标签不仅细化了用户的需求，也为推荐系统提供了更为丰富和精确的输入，从而大大提高了推荐的准确性和个性化程度。

综上所述，本文的研究不仅具有重要的理论意义，通过引入和应用最新的大语言模型技术，推动了推荐系统和自然语言处理领域的研究进展；同时，也具有显著的实践价值，为电商平台提供了一种创新的解决方案，以更好地满足用户的个性化需求，促进电子商务与金融科技领域的健康发展。

2 文献综述

本文的推荐系统流水线结合了多种不同领域的大语言模型框架以及机器学习框架。

2.1 词向量嵌入技术

词向量嵌入技术是自然语言处理（NLP）领域的一项核心技术，它将词语转换为计算机可以理解的数学向量。通过这种方式，模型能够捕捉到词语之间的语义关系，包括相似性和上下文关联性。词向量的引入极大地提升了机器对自然语言的理解能力，是众多 NLP 任务（如文本分类、情感分析、机器翻译等）的基础。

分布式表示（也称为词嵌入）技术能够将词表示为稠密的向量，这些向量在低维空间中。这种表示方法的基本假设是“语境相似的词其语义也相似”，这一假设也被称为分布假说。主要的词嵌入模型包括 Word2Vec、GloVe 和 FastText 等。

2.1.1 Word2Vec

Word2Vec 是由 Google 团队提出的模型，有两种架构：连续词袋模型（CBOW）和跳字模型（Skip-gram）。CBOW 预测目标词基于上下文，而 Skip-gram 则反其道而行之，预测上下文基于目标词。Word2Vec 的目标是使得模型学习到的词向量能够将具有相似上下文的词聚集在向量空间中的相近位置。本文的 Ref2Label 模型采用了 Word2Vec 模型作为其底层词嵌入层。

2.1.2 GloVe

全局向量的词嵌入 (GloVe) 是一种在全局词-词共现矩阵上进行训练的词嵌入方法。它结合了局部上下文窗口方法和全局矩阵分解技术的优点, 目的是更好地捕捉词之间的共现信息, 反映词与词之间的语义关系。

2.1.3 FastText

FastText 是 Facebook 研究团队提出的一种词嵌入方法, 与 Word2Vec 类似, 但它在训练过程中不仅考虑词本身, 还考虑词内的子词 (subword)。这种方法对处理形态丰富的语言 (如德语) 特别有效, 因为它能够捕捉到词缀等语言特性, 提升模型对未见词的处理能力。

2.2 Transformer 架构

Transformer 架构自 2017 年由 Vaswani 等人在论文《Attention is All You Need》中提出以来, 已经成为自然语言处理 (NLP) 领域的一个重要里程碑。该架构摒弃了传统的循环神经网络 (RNN) 和卷积神经网络 (CNN) 模型, 完全基于注意力机制 (Attention Mechanism), 在处理序列数据方面展现出了卓越的性能。Transformer 架构特别适合处理长距离依赖问题, 能够有效捕捉序列内各元素间的全局关系。

2.2.1 编码器 (Encoder)

Transformer 模型的编码器由 N 个相同的层堆叠而成, 每一层都包含两个主要的子层: 一个多头自注意力机制 (Multi-Head Self-Attention Mechanism) 和一个简单的、位置全连接的前馈网络 (Positionwise Fully Connected Feed-Forward Network)。这两个子层均采用了残差连接 (Residual Connection) 和层归一化 (Layer Normalization), 这有助于避免在深层网络中出现的梯度消失或爆炸问题。

多头自注意力机制可以让模型同时关注到输入序列的不同位置, 从而更好地理解序列内的上下文关系。通过将注意力机制分解为多个“头”, 模型可以在不同的表示子空间中捕捉到丰富的信息。

2.2.2 解码器 (Decoder)

与编码器结构类似, 解码器也是由 N 个相同的层堆叠而成, 但在每个解码器层中添加了一个额外的多头注意力子层, 用于处理编码器的输出。这个额外的注意力子层允许解码器关注到编码器的所有位置, 这在进行序列到序列的任务 (如机器翻译) 时尤为重要。

解码器的多头自注意力机制被修改为防止位置向后看 (即在生成第 i 个词时, 只能看到第 i 个词之前的词), 这种机制被称为掩蔽 (Masked) 自注意力机制。

2.2.3 注意力层 (Attention Layer)

Transformer 架构的核心是其注意力层, 特别是多头注意力机制。注意力函数可以描

述为根据查询（Query）和一组键值对（Key-Value Pairs）来计算输出，其中查询、键、值和输出都是向量。输出是值的加权和，权重由查询与所有键的相似度计算得出。

Transformer 采用的是缩放点积注意力（Scaled Dot-Product Attention），其计算效率较高。在多头注意力机制中，模型将查询、键和值通过不同的线性投影分割成多个头，然后每个头独立进行注意力计算，最后将所有头的输出拼接起来，再次进行线性投影以得到最终的输出。这种机制使得模型能够在不同的表示子空间中捕获信息，增强了模型的表达能力。

Transformer 架构通过其创新的设计，有效地解决了序列处理任务中的长距离依赖问题，并极大地提高了训练的效率。它已经成为许多先进 NLP 模型的基础，如 BERT、GPT 系列等，推动了整个自然语言处理领域的进步。

2.3 模型微调(Fine-tuning)

模型微调（Fine-tuning）是一种迁移学习技术，通过在预训练的模型基础上继续训练，以适应特定任务的过程。这种方法在自然语言处理（NLP）领域尤其流行，因为它允许开发者利用预训练模型（如 OpenAI 的 GPT 系列）在大规模数据集上学到的丰富知识和模式，快速地适应并优化特定的下游任务，无论是文本分类、情感分析还是问答系统。

OpenAI 提供了一种方法，允许开发者通过输入特定格式的数据文件来微调其模型，例如 ChatGPT。这一过程通常涉及到使用 JSONL 格式的文件，该文件格式是 JSON 数据的一种变体，每一行是一个独立的 JSON 对象，允许模型以流式传输的方式读取大量数据。在微调 ChatGPT 模型的场景中，每个 JSON 对象通常包含两部分：输入文本和期望的输出文本，模拟了一个交互式对话的环节。

进行模型微调时，开发者需要准备一个或多个 JSONL 文件，这些文件包含了特定任务的示例对话。每个文件中的每一行都代表一个训练样本，样本中包含了对话的输入（例如用户的问题）和模型应该产生的输出（例如正确的回答或反应）。通过这种方式，模型不仅学习到了语言的基本结构，还学习到了如何针对特定任务生成合适的回答或内容。

微调过程中，模型的参数会根据这些特定任务的数据进行调整，以最小化预测输出和实际输出之间的差异。这种细微的调整使得模型更加适应该任务，从而在特定领域的应用中表现得更加出色。微调可以显著提高模型在特定任务上的性能，尤其是在预训练模型未能充分覆盖的领域或特定用例中。

微调的优势在于能够借助预训练模型强大的通用性能力，并通过少量特定领域的数据进行调整，使其更适合特定的应用场景。这种方法节省了从头开始训练模型所需的大量时间和资源，同时保持了模型的高性能和灵活性。

2.4 文本聚类

文本聚类是自然语言处理（NLP）领域中的一个重要任务，它旨在自动将文本集合分组，使得同一组内的文本在内容上更加相似，而不同组之间的文本则相对不同。这一过程不依赖于预先标注的类别信息，是一种无监督学习方法。文本聚类的应用范围广泛，包括信息检索、文档管理、话题发现和内容推荐等。

在实现文本聚类时，文本数据首先需要被转换成机器可处理的数值形式，即词向量嵌入。词向量嵌入技术如 Word2Vec、GloVe 或 FastText 等，能够将单词或短语映射到高维空间中的向量，这些向量捕捉了词汇的语义特征和上下文相似性。通过词向量嵌入，文本数据被有效地转化为数值型特征，为后续的聚类分析提供了基础。

在本项目中，通过将标签转化为词向量，我们可以利用这些向量之间的距离度量来反映标签内容的相似度。随后，采用 k-means 算法对这些词向量进行聚类。k-means 是一种广泛使用的聚类算法，通过迭代寻找 k 个聚类的中心，并将每个点分配到最近的聚类中心，以此来最小化每个点到其聚类中心的距离平方和。在文本聚类的上下文中，这意味着算法会将语义上相似的标签归为同一类。

项目中的一个关键步骤是计算每个聚类中心与词向量之间的距离，以找到离每个聚类中心最近的词向量。这个最近的词向量可以被视为该聚类的“代表”，因为它最能反映该聚类的主要语义。这种方法不仅帮助我们理解每个聚类代表的内容，还能在一定程度上简化标签的管理和搜索，使得相似的标签更易于检索和组织。

文本聚类通过这种方式，实现了对大量文本或标签的有效组织和分类，提高了信息检索的效率和准确性。此外，该方法还能揭示文本数据中潜在的结构和模式，为深入分析和理解数据提供了有价值的洞察。在电商平台等应用场景中，文本聚类技术可以帮助改善商品标签的管理，优化搜索和推荐系统，从而增强用户体验和满意度。

3 技术路线

3.1 开发工具与环境

本项目的技术栈涵盖了现代 Web 开发和机器学习领域的先进技术，具体使用了 Gin 框架和 Vue Vite 框架分别开发电商金融系统的后端和前端。在机器学习模块，采用 Pytorch 和 Transformer 框架进行模型的训练，同时使用 Sklearn 库中的 K-means 算法对结果进行聚类处理。数据存储方面，结构化数据和词向量数据分别存储在 MySQL 和 PostgreSQL 数据库中。此外，本项目还利用了 OpenAI 平台提供的 gpt-3.5turbo 接口进行模型的 fine-tuning 训练以及词向量嵌入，以增强系统的智能化处理能力。

3.1.1 Gin 框架

Gin 是一个用 Go 语言编写的高性能 Web 框架。它提供了一套简洁有效的 API，使得开发者能够快速构建高性能的 Web 应用。Gin 框架的设计哲学是提供最小的、最快的

框架，而不牺牲灵活性。它通过使用 `httprouter` 路由库来实现路由功能，该库优化了路由的匹配过程，从而使 `Gin` 成为处理 `Web` 请求的高效框架。

3.1.2 Vue Vite 框架

`Vue Vite` 是一个现代化的前端开发框架，结合了 `Vue.js` 框架和 `Vite` 构建工具的优势。`Vue.js` 是一个渐进式 `JavaScript` 框架，用于构建用户界面，特别是单页应用（SPA）。`Vite` 作为一个前端构建工具，提供了快速的冷启动、即时模块热更新（HMR）和真正的按需编译。`Vue Vite` 框架通过利用现代浏览器支持的 `ES` 模块导入特性，实现了无需打包操作的开发服务器，大大提高了开发效率。

3.1.3 PyTorch

`PyTorch` 是一个开源的机器学习库，广泛应用于计算机视觉和自然语言处理等领域。它提供了强大的张量计算功能以及动态计算图的特性，使得模型的设计、实验和调试过程更为直观和灵活。`PyTorch` 的核心是提供了一个类似于 `NumPy` 的张量库，但它支持 `GPU` 加速，可以显著提高计算效率。

3.1.4 PostgreSQL

`PostgreSQL` 是一个功能强大的开源对象关系数据库系统，以其可靠性、健壮性和性能而闻名。本文利用 `pgvector` 插件，旨在为数据库提供高效的向量搜索能力。它允许用户在 `PostgreSQL` 中存储和查询向量数据，非常适合于实现词向量等机器学习应用的数据存储需求。`pgvector` 通过优化向量之间的距离计算，支持快速的相似性搜索操作，使其成为处理大规模向量数据的理想选择。

3.2 数据集

本文基于 `Jianmo Ni` 所整理的“`Amazon Review Data (2018)`”数据集，特别关注其中的“`Grocery and Gourmet Food`”部分作为研究对象。该数据集是 2014 年 `Amazon` 评论数据集的更新版本，由加州大学圣迭戈分校的 `Jianmo Ni` 负责维护。本数据集包含了评论、产品元数据（描述、类别信息、价格、品牌和图像特征）。本文的电商系统的商品数据来自数据集中的产品元数据。

在本研究中，我们专注于数据集中的“`Grocery and Gourmet Food`”部分，旨在开发一个名为 `Ref2Label` 的推荐系统。为了训练该系统，我们采用了 `OpenAI` 提供的 `GPT-3.5` `API`，对选定的评论数据块进行处理，请求 `GPT-3.5` 生成参考答案作为模型的训练集。鉴于 `OpenAI` 接口的使用成本较高和响应速度的限制，我们仅对数据集的一小部分数据进行了这样的训练。

```
管理员: C:\Windows\System32\cmd.exe - python main.py
C:\Users\Administrator\OneDrive - VG Enterprise\桌面\py>python main.py
try to get answer from openAI
OpenAI create training set Processing: 1% | 531/60291 [2:44:04<984:30:51, 59.31s/it]
```

图 3-1 调用 OpenAi 接口获得训练集

通过这种方式，我们利用了大规模的用户评论数据和高级的自然语言处理技术，以期提高推荐系统的准确性和用户体验。此外，本研究的数据处理和模型训练方法也为其他领域的研究者提供了一个利用大规模文本数据和深度学习模型进行研究的实例。

3.3 系统功能简介

在本研究中开发的电商金融系统中，前后端系统为用户提供了全面的购物体验，支持用户注册、浏览商品、添加商品到购物车、下单、查看订单以及获取个性化商品推荐等功能。管理端的设计使得商城管理员能够高效地编辑商品信息、商城公告以及配送信息，确保了商城运营的灵活性和高效性。

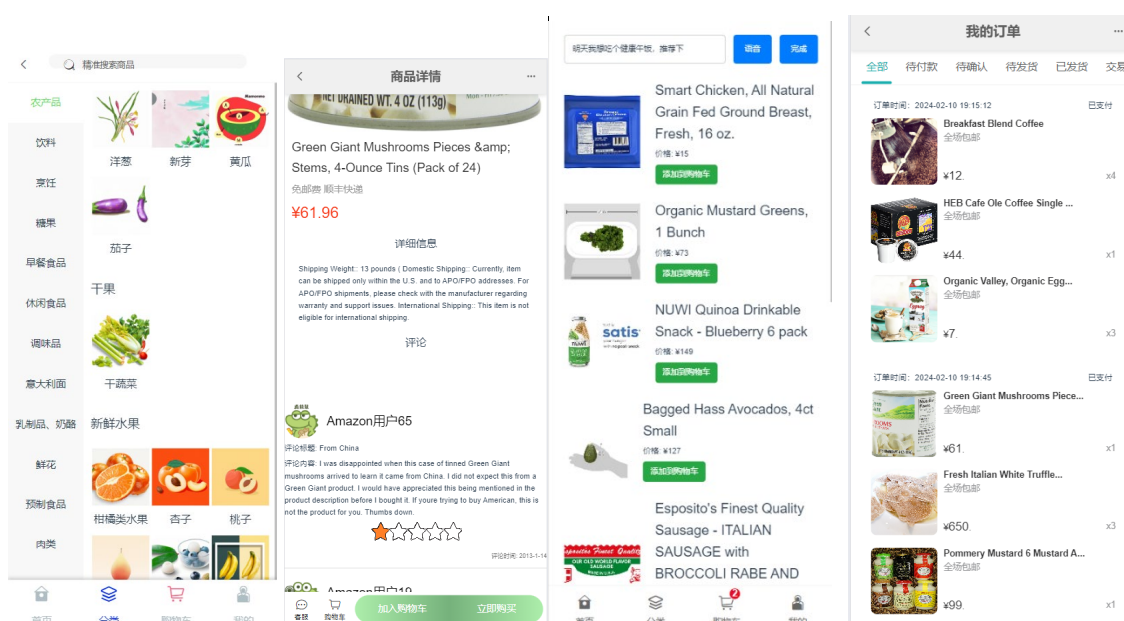


图 3-2 电商金融系统用户端页面展示

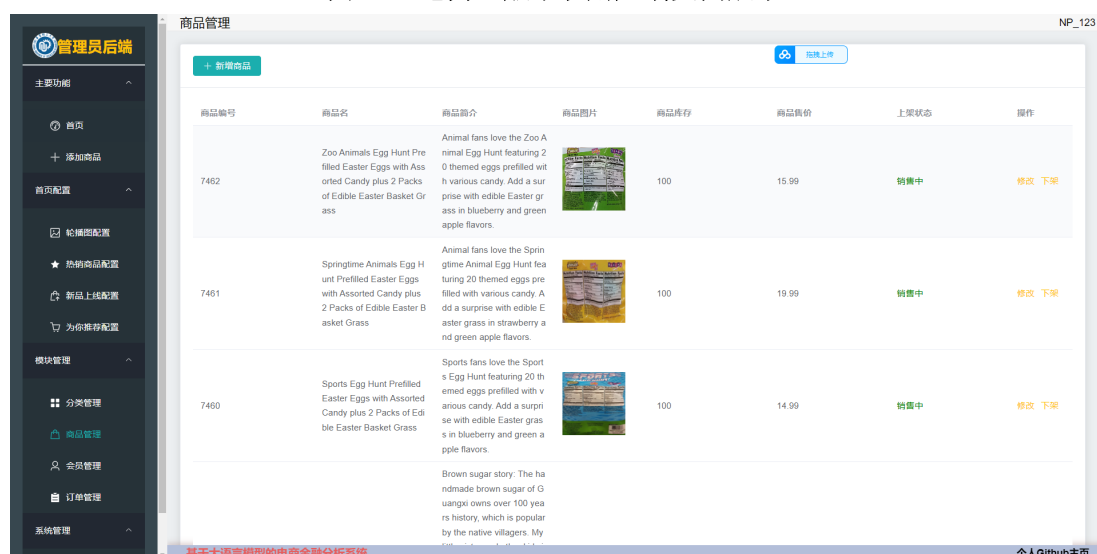


图 3-3 电商金融系统管理端页面展示

深度学习模块的核心是利用 Pytorch 训练的 Ref2Label 的大语言模型，该模型基于

Transformer 架构，结合了 Kmeans 聚类算法，构建了一个能够持续自我更新的智能推荐系统。通过对大量用户评论的深入分析和处理，ref2label 模型能够自动提炼出能够最准确反映商品特性的标签，进而利用这些标签为用户推荐最合适的商品。这一过程不仅显著提升了推荐的准确性，也极大地丰富了用户的购物体验。模型通过细致的 fine-tuning 和词嵌入技术的应用，进一步增强了推荐系统的性能。

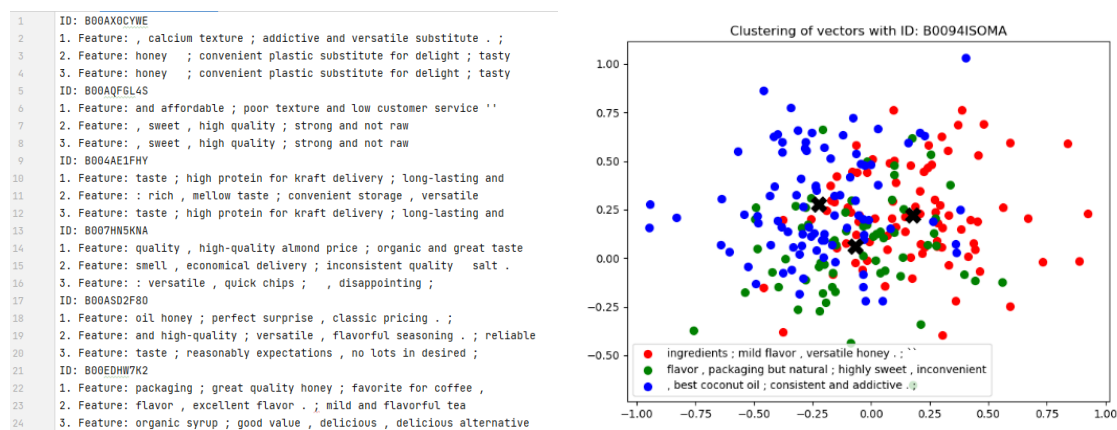


图 3-4 聚类后商品评论标签

3.3.1 用例图

本系统的角色分为用户和管理员，如图 3-5 所示。

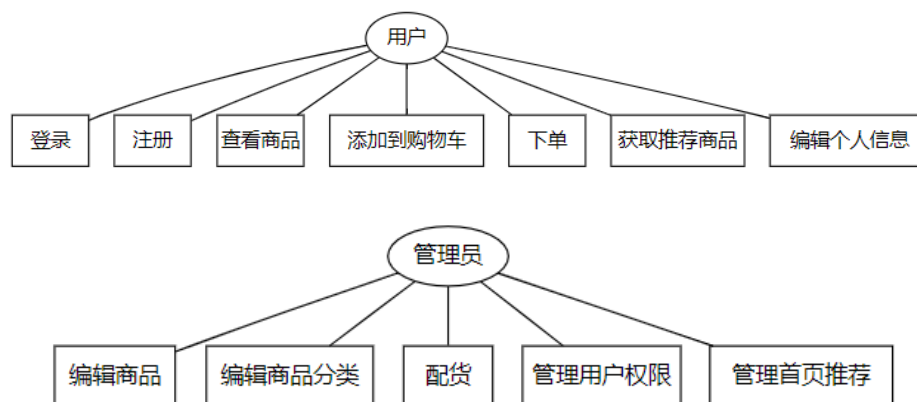


图 3-5 用户以及管理员用例图

注册用户的用例：

1. 用户可以登录、注册系统：用户可以创建一个新的账户或者使用现有的账户信息登录系统。所有用户数据，包括个人信息、登录凭证等，都将安全地存储在服务器的数据库中。
2. 用户可以查看商品：注册用户可以浏览平台上的所有商品。这包括查看商品详情、价格、评价等信息。
3. 用户可以添加商品到购物车：用户可以选择心仪的商品加入到他们的购物车中，以便之后进行购买。

- 4. 用户可以下单：用户可以从购物车中选取商品，确认购买信息后，完成订单的创建。订单信息将被记录并用于后续的配送处理。
- 5. 用户可以获取推荐商品：根据用户的浏览历史和购买习惯，系统会自动推荐相关商品。这一过程可能涉及到深度学习或其他智能算法，以提高推荐的准确性。
- 6. 用户可以编辑个人信息：用户可以更新他们的个人信息，包括但不限于联系方式、配送地址等，确保订单能够正确配送到达。

管理员的用例：

- 1. 管理员可以编辑商品：管理员有权限对商品信息进行编辑，包括更新商品描述、价格、库存量等信息。
- 2. 管理员可以编辑商品分类：管理员可以添加、修改或删除商品分类，以便更好地组织平台上的商品，帮助用户更容易地找到他们想要的商品。
- 3. 管理员可以对已有的订单进行配货：管理员负责处理订单，包括确认订单信息、准备货物、安排发货等，确保客户能够及时收到商品。
- 4. 管理员可以管理用户权限：管理员可以对用户的权限进行管理，包括限制或恢复用户的购买权限，确保平台的正常运营。
- 5. 管理员可以管理商城首页推荐：管理员可以选择特定的商品或促销信息展示在商城首页，吸引用户关注和增加销量。

3.3.2 系统功能模块简介

在本研究中开发的电商金融系统是一个综合性的电子商务解决方案，它集成了前端系统、管理端功能以及先进的深度学习技术，旨在为用户和管理员提供高效、个性化的服务。以下是系统的主要功能模块简介。

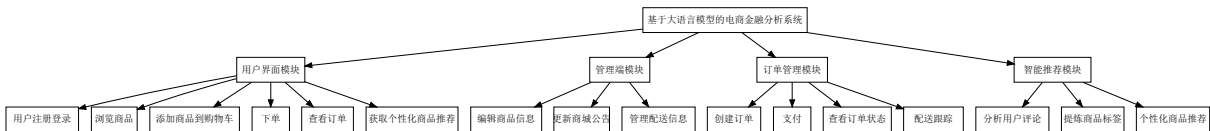


图 3-6 系统模块示意图

3.3.3 用户界面模块

该模块是用户与电商平台互动的前端界面，支持用户进行注册、登录以及浏览商品。用户可以通过简洁直观的界面添加商品到购物车、下单以及查看订单状态。此外，用户还能接收到基于个性化算法生成的商品推荐，这些推荐会显示在用户界面的显著位置，以提高用户的购物体验和销售效率。

3.3.4 管理端模块

为了确保电商平台的灵活性和高效性，管理端模块为商城管理员提供了一套全面的管理工具。管理员通过这个模块可以高效地编辑商品信息、更新商城公告以及管理配送

信息。这个模块的设计考虑到了操作的便捷性和信息管理的安全性，使得管理员能够轻松地维护平台内容，保证信息的准确性和时效性。

3.3.5 订单管理模块

订单管理模块是处理用户购物流程的核心，它支持订单的创建、支付、查看以及配送跟踪。该模块能够处理用户从添加商品到购物车到完成购买的整个过程，包括订单状态更新、支付验证以及配送安排。通过这个模块，用户可以实时跟踪订单状态，而管理员也可以高效地管理订单流程，确保用户满意度。

3.3.6 智能推荐模块

智能推荐模块是该系统的技术亮点，它基于 Pytorch 框架和 Transformer 架构开发的 Ref2Label 大语言模型。该模型结合 Kmeans 聚类算法，通过深入分析和处理大量用户评论，自动提炼出准确反映商品特性的标签。利用这些标签，智能推荐系统能够为用户推荐最合适的商品。通过细致的 fine-tuning 和词嵌入技术的应用，该模块不仅提升了推荐的准确性，也极大地丰富了用户的购物体验。

3.4 Seq2Label 模型

本项目的核心模块是一个基于深度学习的推荐系统，名为 Ref2Label，其设计目的在于将用户评论高效转化为商品标签，以提升商品推荐的准确性和用户体验。Ref2Label 模型采用了先进的 Transformer 架构和 Kmeans 聚类算法，结合了用户评论数据和商品特性，通过深度学习技术自动生成与商品相关的标签。这些标签不仅精准反映了商品的核心特点，也贴合用户的真实需求和偏好。

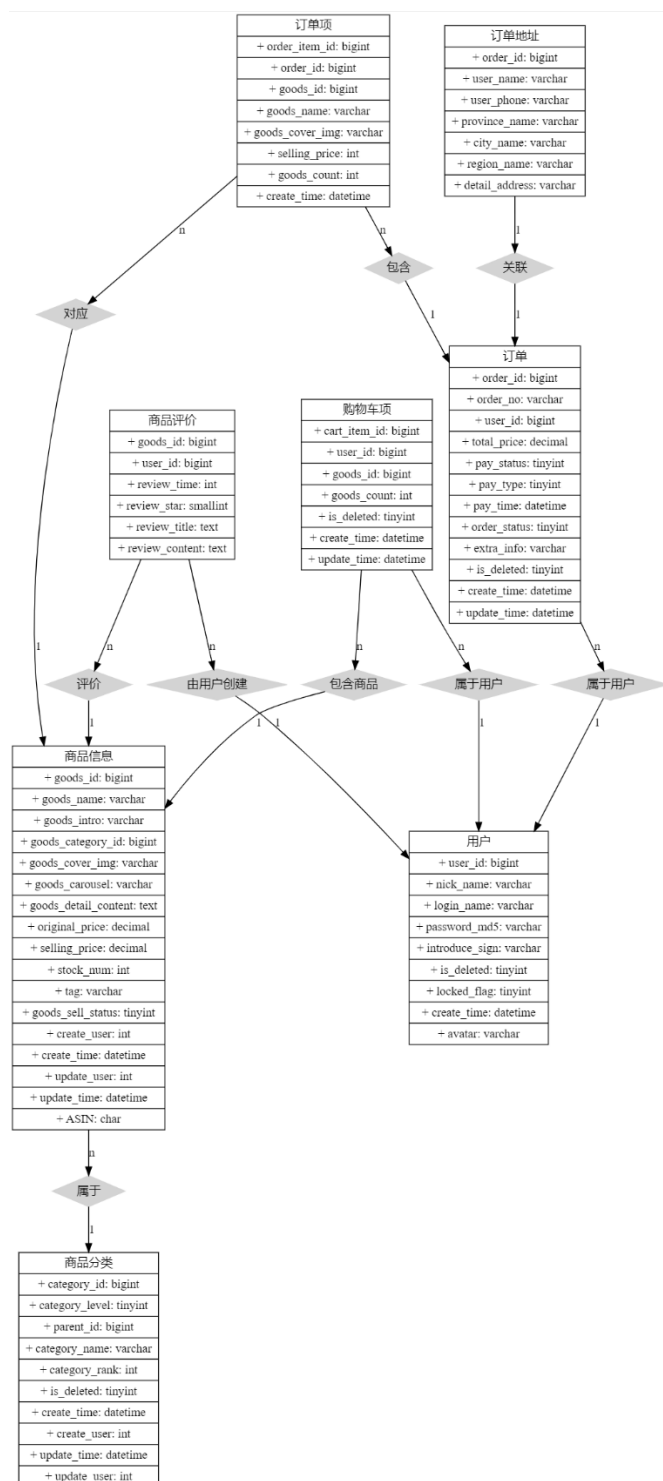
在模型训练过程中，我们采用了编码器-解码器（Encoder-Decoder）框架，其中包括两个独立的编码器（encoder1 和 encoder2）用于处理不同类型的输入数据，并通过一个解码器（decoder）生成最终的标签。模型的训练涉及到多个步骤，包括输入变量的预处理、模型参数的优化、损失函数的计算以及梯度的反向传播等。此外，我们还引入了 teacher forcing 策略来加速模型的收敛速度，并使用了梯度裁剪（Gradient Clipping）技术来防止梯度爆炸问题。为了提高模型的泛化能力，我们在训练过程中采用了动态学习率调整机制，以及在验证集上进行早停（Early Stopping）以避免过拟合。

此外，Ref2Label 模型的训练和验证过程中，我们充分利用了 CUDA 技术来加速计算，以应对大规模数据集的处理需求。通过这一系列精心设计和优化的训练流程，Ref2Label 模型在实验中展现出了优异的性能，有效地提升了推荐系统的准确率和用户满意度，证明了该模型在电商推荐系统中的应用价值和潜力。

3.5 数据库设计

3.5.1 E-R 图

如图 3-7 所示,通过实体集、属性和联系集的图形化表示,展示了数据库的结构。本系统包含了商品分类、商品信息、商品评价、订单、订单地址、订单项、购物车项以及用户等实体集,这些实体集通过一系列的联系集相互连接,形成了电商金融系统的数据模型基础。



商品分类实体集：分类 ID、分类级别、父分类 ID、分类名称、排序值、删除标识、创建时间、创建者 ID、修改时间、修改者 ID。其中分类 ID 是自增主键，分类名称是唯一键，父分类 ID 用于表示层级关系。

商品信息实体集：商品 ID、商品名称、商品简介、关联分类 ID、商品主图、商品轮播图、商品详情、原始价格、销售价格、库存数量、商品标签、商品销售状态、添加者 ID、添加时间、修改者 ID、修改时间、ASIN。其中商品 ID 是自增主键，添加时间为商品创建时间，商品主图和轮播图为商品图片存储于服务器中的地址。

用户实体集：用户 ID、昵称、登录名、密码、个性签名、注销标识、锁定标识、注册时间、头像。其中用户 ID 是自增主键，登录名是唯一键。

订单实体集：订单 ID、订单号、用户 ID、总价、支付状态、支付类型、支付时间、订单状态、额外信息、删除标识、创建时间、更新时间。其中订单 ID 是自增主键，订单号是唯一键。

订单地址实体集：订单 ID、收货人姓名、收货人手机号、省、市、区、详细地址。其中订单 ID 是主键，同时也是外键，关联到订单实体。

订单项实体集：订单项 ID、订单 ID、商品 ID、商品名称（快照）、商品主图（快照）、销售价格（快照）、商品数量（快照）、创建时间。其中订单项 ID 是自增主键，订单 ID 和商品 ID 为外键，分别关联到订单实体和商品信息实体。

购物车项实体集：购物车项 ID、用户 ID、商品 ID、商品数量、删除标识、创建时间、更新时间。其中购物车项 ID 是自增主键，用户 ID 和商品 ID 为外键，分别关联到用户实体和商品信息实体。

3.5.3 联系集及其属性

属于联系集（商品信息与商品分类）：一对多联系集，表示一个商品分类可以包含多个商品信息，但每个商品信息只属于一个商品分类。

评价联系集（商品评价与商品信息）：一对多联系集，表示一个商品信息可以有多个商品评价，但每个商品评价只能评价一个商品信息。

拥有联系集（订单与用户）：一对多联系集，表示一个用户可以拥有多个订单，但每个订单只属于一个用户。

关联联系集（订单地址与订单）：一对一联系集，表示每个订单有一个具体的订单地址。

包含联系集（订单项与订单）：一对多联系集，表示一个订单可以包含多个订单项，但每个订单项只属于一个订单。

对应联系集（订单项与商品信息）：一对一联系集，表示每个订单项对应一个具体的商品信息。

属于用户联系集（购物车项与用户）：一对多联系集，表示一个用户可以拥有多个购

物车项，但每个购物车项只属于一个用户。

包含商品联系集（购物车项与商品信息）：一对一联系集，表示每个购物车项包含一个具体的商品信息

3.6 项目成本估算

估计项目成本如表 3.1 所示。

表 3.1 项目成本预算表

序号	项目	工作量(天)	费用(元)
1	租用云服务器(1 年)	-	2000
2	调用 OpenAI 接口获取数据集	10	200
3	需求分析	5	100
4	深度学习框架调试	30	500
5	后端开发	20	800
6	前端开发	10	600
7	系统调试	3	400
8	测试	2	300
9	部署	1	200
	总计	81	4700

4 进度安排

进度安排如表 3.2 所示。

表 3.2 进度安排细则

序号	各阶段工作	起止日期
1	选题并查阅相关文献，完成文献翻译和开题报告。	2023.12 - 2024.1
2	熟悉开发工具以及相关技术，包括深度学习框架和云服务器操作。	2024.1 - 2024.2
3	开发深度学习模块，包括算法研究和模型训练。	2024.2 - 2024.3
4	开发后端逻辑，包括商品信息管理、订单处理等功能。	2024.3 - 2024.4
5	开发前端界面和用户交互功能。	2024.4 - 2024.5
6	系统测试，包括单元测试、集成测试和性能测试。	2024.5 - 2024.5
7	系统调试和优化，确保系统稳定运行。	2024.5- 2024.6
8	撰写毕业论文，总结项目开发过程和成果。	2024.5 - 2024.6
9	整理并提交毕设设计相关材料，制作论文答辩 PPT，进行毕业设计答辩。	2024.6

5 参考文献

[序号] 著录内容（首行缩进两格，小四号字，

